Remarks

Claim 1 has been amended to be more narrow. Just "display regions for graphical representations of access control settings for the resource" was less specific than now requiring "display regions for graphical representations of all access control settings for the resource", with the former accommodating omission of important settings.

Claim 7 has been amended to be more narrow. Just "a set of groups, users and roles and their respective access privileges as defined by existing structured data for the resource" was less specific than now requiring "the set of groups, users and roles defined by existing structured data for the resource and their respective access privileges", with the former accommodating omission of groups, users and roles that are allowed to access the resource.

Claim 7 has been amended to correct a minor oversight. With a "set of groups, users and roles" being mentioned then instead of "the result of transforming the set of groups and users" it also should be "the result of transforming the set of groups, users and roles".

Claim 10 has been amended to be more narrow. Now it requires that "the resource is a digital document".

Claim 43 is more narrow than claim 38. It adds the same restrictions as claim 9.

In response to action item 4, regarding claim 1:

Execution versus access; and no representation of content of a digital document.

The thrust of Hayes is that an administrator can control execution of compiled code, i.e. of applets. In contrast, the present invention concerns itself with a user controlling access to data, a document he is working on.

There has been a tradition in the art of distinguishing between code and data, between an executable program and a document that a user is working on.

There has been a tradition in the art of distinguishing between permissions to access information versus permission to execute code. E.g. Unix file system read versus execute permissions. Hayes teaches control over execution of programs, not control of access per document.

The resources which Hayes writes about aren't digital documents, but "applets". An applet is executable, complied code.

Consequentially there is an important part of claim 1 that is not shown or taught by Hayes:

Hayes doesn't show normal size, legibly scaled, unabridged representation of the content of the resource wherein the resource is a digital document. In Hayes there is no indication of concurrently showing access control settings and content, data or documents, let alone them being concurrently operable.

In the spirit of the current invention one would expect to control access while seeing the applet showing its applet interface, e.g. a 3D model of a molecule shown by the applet. Even then with closer resemblance in superficial appearances, even then rights to execute code would be a different concept than rights to access data. "Who is allowed to use the word processing program?" of Hayes versus "Who is allowed to read this document?" of the current invention.

If Hayes fails to anticipate essential features of claim 1 then Hayes hasn't anticipated any claim dependent on claim 1.

Applicant's amendment filed 2008-06-11 in response to concerns by Examiner specifically has narrowed claim 1 to "wherein the resource is a digital document".

Never shows for an applet who all is allowed to execute it.

When following Hayes Figs. 13-24 one first sees the group at the center of navigation. For a group there is a list of applets permitted or denied. For a group there is a list of users.

Then one sees the user at the center of navigation. For a user there is a list of groups, and for a user there is a list of applets.

What is missing is for the applet a list of what groups are allowed to execute it, or what users are allowed to execute it.

If the applet is the resource, what is missing is a representation of all execution control settings for the resource.

Hayes would require a lot of clicking through lists and more UI elements if one wants to check all groups and users in order to find out the complete set of who is allowed to execute an applet, let alone a lot of memorizing.

If Hayes fails to anticipate essential features of claim 1 then Hayes hasn't anticipated any claim dependent on claim 1.

In response to action item 4, regarding claims 5, 36 and 38:

Applicant's amendment filed 2008-04-07 defines: The term "likeness of a person" means a identifying pictorial representation of the person, an imitative image, e.g. an identifying photograph, possibly a modified photograph or a machine processed image of that person that sufficiently corresponds to the person's appearance to allow a normally skilled human to identify the person in an encounter with normal visual contact.

There is no such concept in Hayes. There are no photos of users. There is no use of photos in admin UI for access control.

If Hayes fails to anticipate essential features of claim 5 then Hayes hasn't anticipated any claim dependent on claim 5.

In response to action item 4, regarding claim 7:

Hayes doesn't show the result of transforming the set of groups, users and roles defined by existing structured data for the resource and their respective access privileges into a corresponding set of individual users and their respective effective access privileges.

Even if it would perform any such transformation during access control functionality at runtime — Hayes doesn't disclose it and performing such transformation doesn't appear to be desirable for an efficient implementation — relevant to claim 7 Hayes definitely doesn't teach showing its results in the UI.

If Hayes fails to anticipate essential features of claim 7 then Hayes hasn't anticipated any claim dependent on claim 7.

In response to action item 5:

Steinberg apparently has been introduced as a reference because it teaches use of access control for the administrator of a machine that processes photos, which has little to do with using photos in access control. Nevertheless we can discuss the differences in image processing.

Steinberg concerns itself with image correction to correct camera variations. Central to Steinberg is taking a picture with a camera of a test sheet and from then on knowing what that specific camera's variation is. All pictures coming from that camera from then on will be corrected by compensating for that specific camera's variation, essentially by shifting image values into the opposite direction of the camera's "deficiency" in order to get closer to the true colors of objects.

In contrast, the present invention tries to make all pictures the same, like a lawn of equally tall equally green grass. The present invention doesn't want to know about camera properties. All id photographs are supposed to be made more equal.

In response to action item 7, regarding claim 10:

Sekiguchi doesn't teach user interface showing log information. Sekiguchi uses log information to run algorithms over them. Sekiguchi teaches the use of algorithms to automatically detect leaks of authentication information.

Sekiguchi doesn't teach user interface showing information for one document only.

Hildebrand teaches access control for folders only, and doesn't provide for access control per document. Hence Fig. 5B.1 is for a folder, not for a document.

Hildebrand teaches access control setup by system administrators only, and doesn't provide for access control by a registered user for his own documents.

Hildebrand writing lacks motivation to enable users to see for a single document combined its access control information with its log information.

Such shortcomings make it a far stretch to claim obviousness to reach the present invention.

One should see Hildebrand and Sekiguchi, for whatever reasons of their circumstances, as extremely limited or lacking in providing user interface for log information for professional non-technical users of document editing and storage systems.

One should not expect user interface innovation coming from practitioners that are occupied with the workings of granting access, unless they are explicitly teaching it.

Hildebrand is an example of user interface for access control that would be difficult to use for busy professional users, technical or non-technical.

In response to action item 7, regarding claim 10:

Hildebrand Fig. 5B.1 specifically shows a group, which is not an individual user. There is a lack of prior art in access control showing the conversion of access control settings to sets of individual users. Prevalent operations in the art include checking whether a specific user is authorized, e.g. testing membership, and listing the top level components of a setting, e.g. explicitly listing groups and users.

In response to action item 7, regarding claim 28:

Sekiguchi doesn't teach a user interface showing graphical representations of users sorted.

In response to action item 8, regarding claim 12:

Hayes doesn't show normal size, legibly scaled, unabridged representation of the content of documents in the same UI as access control.

Overall view:

User centric.

The present invention offers clear advantages for non-technical or older professionals needing to comply with regulations like HIPAA or Sarbanes-Oxley while manipulating documents, or wanting to obey and enforce other organizational confidentiality requirements: One view, all information. No clicking necessary. Purpose driven, compliance. Putting the user in control without requiring an administrator.

Separate disciplines.

The engineering of servers, which are the hubs of access control, so far apparently hasn't led to efficient user interface for access control. This situation has left professional and non-technical users without practical access control. The present invention improves upon this situation by reaching across not so well connected disciplines.

Respectfully submitted,

/Leo Baschy/

Leo Baschy

Applicant Pro Se

Date 2008-10-16